



Développement d'application statistique process

BUT SD 2ème année - Parcours VCOD

Année 2024 - 2025

Etudiant :
Loïck PERROT

Maitre de stage :
Jonathan BARJOU

Enseignant tuteur :
Nicolas CHAUCHAT

Remerciements

Je tiens à exprimer ma profonde gratitude envers toutes les personnes qui m'ont chaleureusement accueilli au sein de l'entreprise, créant un environnement propice à l'apprentissage et à l'épanouissement professionnel. Je remercie tout particulièrement M. Barjou et M. Descomps, pour leur bienveillance constante, leur grande disponibilité, ainsi que pour les conseils avisés et l'encadrement de qualité dont j'ai pu bénéficier tout au long de mon stage. Leur accompagnement attentif a été pour moi une source d'enrichissement tant sur le plan technique que sur le plan humain.

Je souhaite également remercier l'IUT pour m'avoir offert cette précieuse opportunité d'effectuer un stage en milieu professionnel, me permettant ainsi de confronter mes acquis théoriques à la réalité du terrain. Cette expérience m'a permis de développer de nouvelles compétences, de gagner en autonomie, et de mieux appréhender les exigences du monde du travail.

Enfin, je tiens à adresser mes sincères remerciements à l'ensemble de l'équipe pédagogique de l'IUT pour leur engagement et leur accompagnement tout au long de l'année. Je remercie tout particulièrement M. Chauchat, mon enseignant référent durant le stage, pour sa disponibilité, ses encouragements, et les conseils pertinents qu'il m'a prodigués. Son soutien m'a été d'une grande aide pour progresser, prendre du recul sur ma mission, et tirer pleinement parti de cette première immersion professionnelle.

A handwritten cursive signature of the word "Merci" in black ink. The signature is fluid and elegant, with the "M" having a long, sweeping initial stroke.

Table des matières :

Table des illustrations :	4
Résumé	5
En français :	5
En anglais :	5
I. Introduction	6
II. Contexte et objectifs	6
1. Description de l'entreprise	6
2. Service	7
3. Présentation du sujet	8
4. Equipe	9
5. Langages et environnement de développement	9
III. Travail réalisé	10
1. Analyse de l'existant	10
2. Récupération des données en SQL	13
3. Développement du back-end (PHP) et paramétrisation des requêtes	15
4. Conception de l'interface utilisateur (front-end)	16
5. Présentation du résultat	21
Conclusion	26
Bilan personnel	26
Glossaire	27
Table des annexes :	29

Table des illustrations :

Figure 1 Plan de l'entreprise Telerad	7
Figure 2 Organigramme du service Méthode et Industrialisation	8
Figure 3 Description de la provenance des données.	11
Figure 4 Exemple d'une partie de l'applicatif attendu.	12
Figure 5 Ancienne requête	14
Figure 6 Nouvelle requête	14
Figure 7 Intégration du calcul des Cp/Cpk dans mon code	16
Figure 8 Définition des variables globales	17
Figure 9 Exemple de params.append et de fetch.....	17
Figure 10 Exemple d'un EventListener	18
Figure 11 Création d'un graphique et insertion des valeurs.....	20
Figure 12 Modification des options et axes du graphique	21
Figure 13 Tableau FPY	22
Figure 14 Graphique FPY	22
Figure 15 Tableau de résultats avec calculs des Cp/Cpk.....	23
Figure 16 Droite de Henry et distribution de loi normale	24
Figure 17 graphique des résultats et courbes Cp/Cpk	25

Résumé

En français :

L'objectif principal de ce stage était de développer un site web interne à l'entreprise permettant d'interroger la base de données Controle_process afin de générer automatiquement des tableaux statistiques et des graphiques, à partir de différents filtres définis par l'utilisateur.

Pour répondre à cette mission, j'ai conçu une application web en utilisant les langages HTML, CSS et JavaScript pour l'interface utilisateur, associés à des scripts PHP assurant la connexion à la base de données et l'exécution des requêtes SQL nécessaires à l'extraction et au traitement des données.

Pour me permettre d'écrire et de tester mon code dans de bonnes conditions, j'ai utilisé différents logiciels, tels que VS Code pour l'édition de mon code, WampServer pour tester mon code en local et MySQL Workbench pour l'écriture des requêtes.

En anglais :

The goal of this internship was to develop an internal website for the company that queries the Controle_process database in order to generate statistical tables and charts based on several filters selected by the user.

To achieve this, I developed the website using HTML, CSS, and JavaScript, connected to PHP scripts responsible for accessing the database and executing the necessary queries.

I. Introduction

Ce stage a été réalisé dans le cadre de ma deuxième année de BUT Science des Données à l'IUT de Pau, du 7 avril au 13 juin 2025. Il avait pour objectif de me confronter à un environnement professionnel concret et de mettre en pratique les compétences acquises durant ma formation, notamment en développement web et en extraction de données depuis une base de données.

J'ai intégré l'entreprise TELERAD, une société spécialisée dans la création d'outils de communications pour les tours de contrôle des aéroports.

Ce rapport présentera dans un premier temps le contexte du stage, les technologies utilisées puis la mission qui m'a été confiée avant de terminer par un bilan personnel de cette expérience.

II. Contexte et objectifs

1. Description de l'entreprise

TELERAD est une PME française de 82 employés spécialisée dans la conception, la fabrication et la commercialisation de systèmes de radiocommunication pour les secteurs aéronautique et maritime, tant civils que militaires. Elle propose une gamme complète de radios VHF, UHF et V/UHF, ainsi que des équipements associés tels que les balises non directionnelles (NDB) et les systèmes d'atterrissement différentiel basés au sol (GBAS). Ils sont présents dans plus de 70 pays au travers de 5 continents. Au 31 décembre 2023 le chiffre d'affaires de l'entreprise était de 13 415 525 €. La figure 1 est un plan du site, le bureau dans lequel je me trouvais pendant le stage se trouve sur le point n°35.



Figure 1 Plan de l'entreprise Telerad

2. Service

Pour ce stage j'opérais au sein du service industrialisation et méthode, ce service a pour but de faire le lien entre la conception (le bureau d'études) et la production. Les missions principales sont :

- Préparer la production :

Transformer un prototype ou un produit conçu par le bureau d'études en un produit fabricable en série de manière fiable, économique et conforme aux exigences qualité, coût, délais (QCD).

- Optimiser les processus de fabrication :

Analyser, définir et améliorer les modes opératoires pour garantir une production efficace et répétable.

- Choix des outils et des moyens :

Sélectionner ou concevoir les outillages, machines, gabarits, bancs de tests nécessaires à la fabrication. Et concevoir des bancs de tests communicants

- Rédaction de la documentation technique :

Elaborer les gammes de fabrication, fiches d'instruction, procédures de test, etc.

- Support à la production :

Aider les opérateurs lors du lancement de nouveaux produits, résoudre des problèmes techniques, faire de la formation si nécessaire.

ORGANIGRAMME SERVICE MÉTHODE ET INDUSTRIALISATION

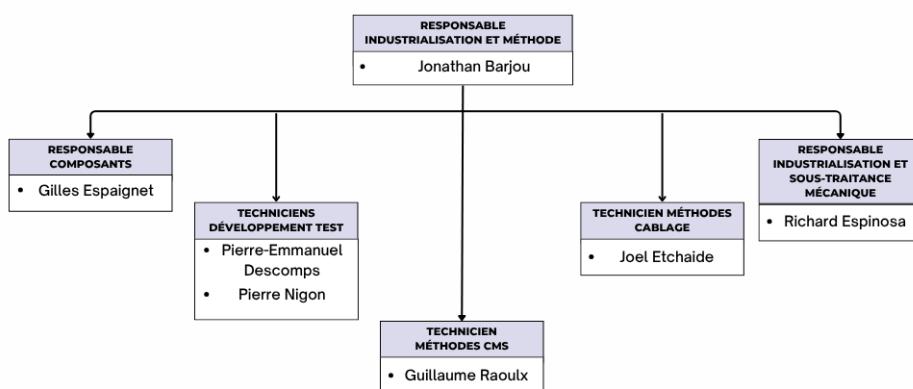


Figure 2 Organigramme du service Méthode et Industrialisation

3. Présentation du sujet

Ma mission consistait à créer un site internet simple d'utilisation permettant de récupérer les données issues des bancs de test et d'en produire des analyses statistiques exploitables par le bureau d'études et le personnel de production.

Ce projet s'inscrivait dans la volonté de répondre à la problématique suivante :

Comment automatiser l'extraction et la visualisation de données issues des bancs de test afin d'en améliorer l'analyse statistique et la prise de décision en milieu industriel ?

Pour mener à bien cette mission, j'ai adopté une démarche organisée en plusieurs étapes, en découplant le projet global en sous-projets successifs.

Dans un premier temps, j'ai pris connaissance de l'ensemble des documents mis à ma disposition, notamment la documentation technique relative à la base de données Controle_process, afin de bien comprendre la structure des données à exploiter.

J'ai ensuite rédigé les requêtes SQL nécessaires pour extraire les données pertinentes en fonction des filtres sélectionnés par l'utilisateur.

Une fois ces requêtes validées, j'ai entamé le développement du back-end de l'application à l'aide de fichiers PHP, en charge de la connexion à la base de données et de l'exécution des requêtes.

Enfin, j'ai développé le front-end du site en HTML, CSS et JavaScript, en veillant à proposer une interface ergonomique, claire et facile à utiliser.

4. Equipe

J'ai mené ce projet de manière autonome, tout en étant encadré par M. Jonathan Barjou et M. Pierre-Emmanuel Descomps.

M. Barjou supervisait principalement la cohérence des traitements et analyses statistiques réalisés, afin de garantir la fiabilité des résultats proposés au bureau d'études.

M. Descomps, quant à lui, m'a accompagné sur les aspects techniques du projet, notamment en ce qui concerne la structure de la base de données et les échanges avec le système d'information.

5. Langages et environnement de développement

Afin de mener à bien cette mission, j'ai dû mobiliser plusieurs outils et langages de programmation. L'objectif étant de développer un site internet, il m'a d'abord fallu mettre en place un environnement de développement local. Pour cela, j'ai utilisé WAMP Server, qui m'a permis de simuler un serveur web en local et de tester efficacement l'application tout au long de son développement.

Pour l'écriture du code, j'ai utilisé Visual Studio Code comme éditeur. Cet environnement m'a permis de développer en HTML, CSS et JavaScript pour la partie front-end, et en PHP pour la gestion des échanges avec la base de données côté back-end.

Étant donné que le site devait interagir avec une base de données, j'ai également eu recours au langage SQL pour rédiger les requêtes nécessaires. Afin de les concevoir, de les tester et de les affiner avant leur intégration dans le code PHP, j'ai utilisé le logiciel MySQL Workbench, qui offre une interface claire et efficace pour interroger et visualiser les données.

L'ensemble de ces outils m'a permis de concevoir une application fonctionnelle, interactive et adaptée aux besoins concrets du service.

III. Travail réalisé

1. Analyse de l'existant

À mon arrivée, plusieurs documents m'ont été remis afin de me présenter les objectifs de la mission ainsi que les outils à ma disposition.

J'ai d'abord reçu un document PowerPoint détaillant les attentes du projet, la provenance des données, ainsi que leur mode de stockage. Comme illustré à la figure 3, la base de données controle_process, sur laquelle reposait l'essentiel de mon travail, centralise les informations issues de différents bancs de test utilisés dans l'entreprise.

Jusqu'à présent, les utilisateurs s'appuyaient sur le logiciel R pour effectuer leurs calculs statistiques et requêtes, ce qui s'avérait souvent fastidieux, notamment pour les utilisateurs non spécialisés. C'est dans ce contexte que mon stage a été défini : concevoir une application web capable de récupérer automatiquement les données de la base, de les traiter de manière efficace, et de les présenter sous forme de tableaux et de graphiques interactifs, facilitant ainsi leur exploitation par le bureau d'études.

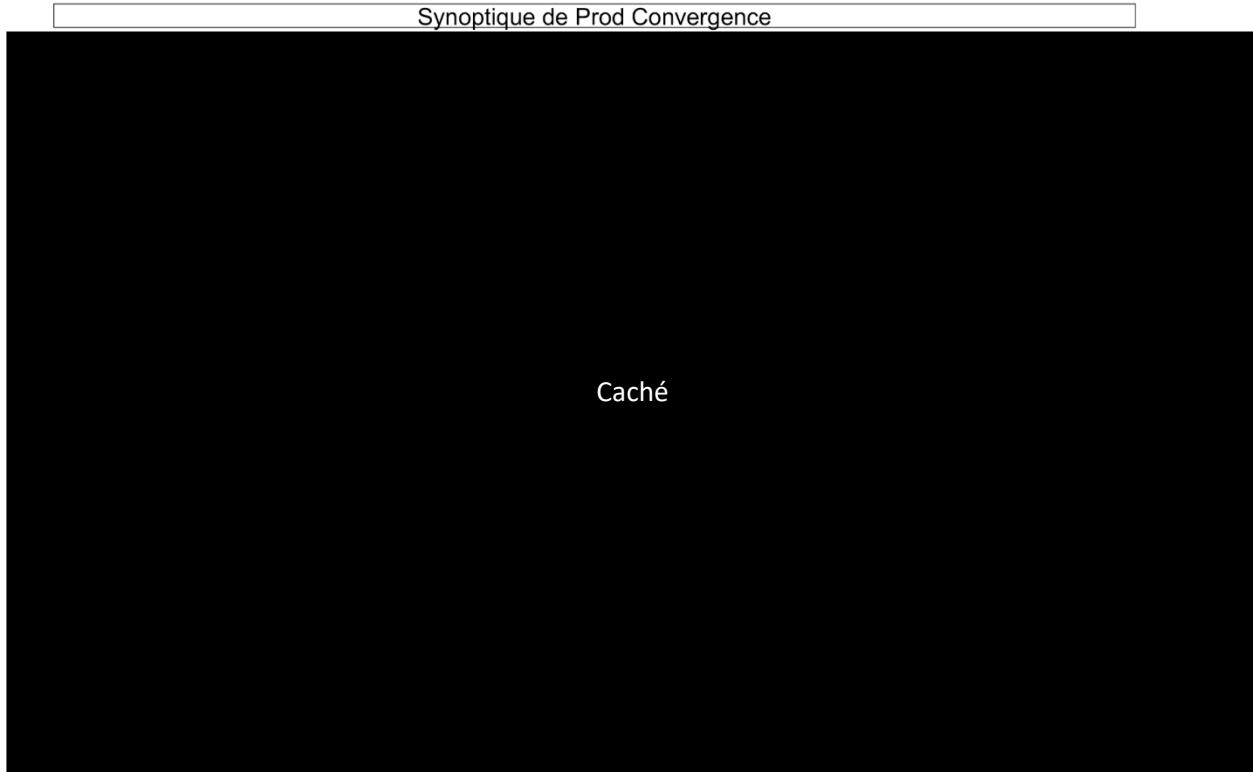


Figure 3 Description de la provenance des données.

Sur la figure 3, on peut voir que plusieurs bancs de mesure collectent des données. Chacun de ces bancs dispose d'un applicatif chargé d'enregistrer les données, puis les personnes qui gèrent ces bancs intègrent ces données dans la base de données. Les bancs en violet représentent ceux qu'ils prévoient de créer. Les bancs en jaune sont déjà existants, tandis que ceux en vert ne transmettent pas de données. Chaque banc vérifie une ou plusieurs caractéristiques des cartes qui sont testées. Les cadenas symbolisent le système d'interblocage, qui empêche de passer à l'étape suivante du test tant que les données de l'étape précédente ne sont pas validées.

Dans le PowerPoint qui m'a été transmis, un fichier Word était intégré en pièce jointe. Ce document contenait une description détaillée de la base de données, de son utilité, des différentes tables ainsi que des champs qui les composent. Il m'a été d'une grande aide pour la construction des requêtes SQL, en me permettant d'identifier précisément les champs à interroger et leur position dans la structure de la base. Nous trouverons des extraits de ce documents dans les annexes 1 et 2.

De plus, un script R m'a été fourni, contenant le code jusqu'alors utilisé par l'entreprise pour la génération des graphiques. Cela m'a permis d'analyser les traitements déjà en place et d'en extraire les fonctions pertinentes, notamment celles nécessaires au calcul des indicateurs statistiques tels que le Cp et le Cpk. Grâce à cette base

existante, j'ai pu mieux comprendre la logique des calculs attendus et les adapter dans le cadre de mon développement en PHP.

En complément, un document Excel m'a également été fourni. Il présentait un exemple du résultat attendu ainsi que plusieurs formules de calcul pouvant m'être utiles pour le traitement des données. Ce fichier a joué un rôle clé dans la conception de mon application : il m'a permis de mieux visualiser les attentes du commanditaire, d'anticiper les traitements statistiques à implémenter et de définir l'interface utilisateur du site. Il m'a ainsi servi de référence pour concevoir à la fois le front-end et le back-end du projet.

Un aperçu de ce document est présenté à la figure 4.

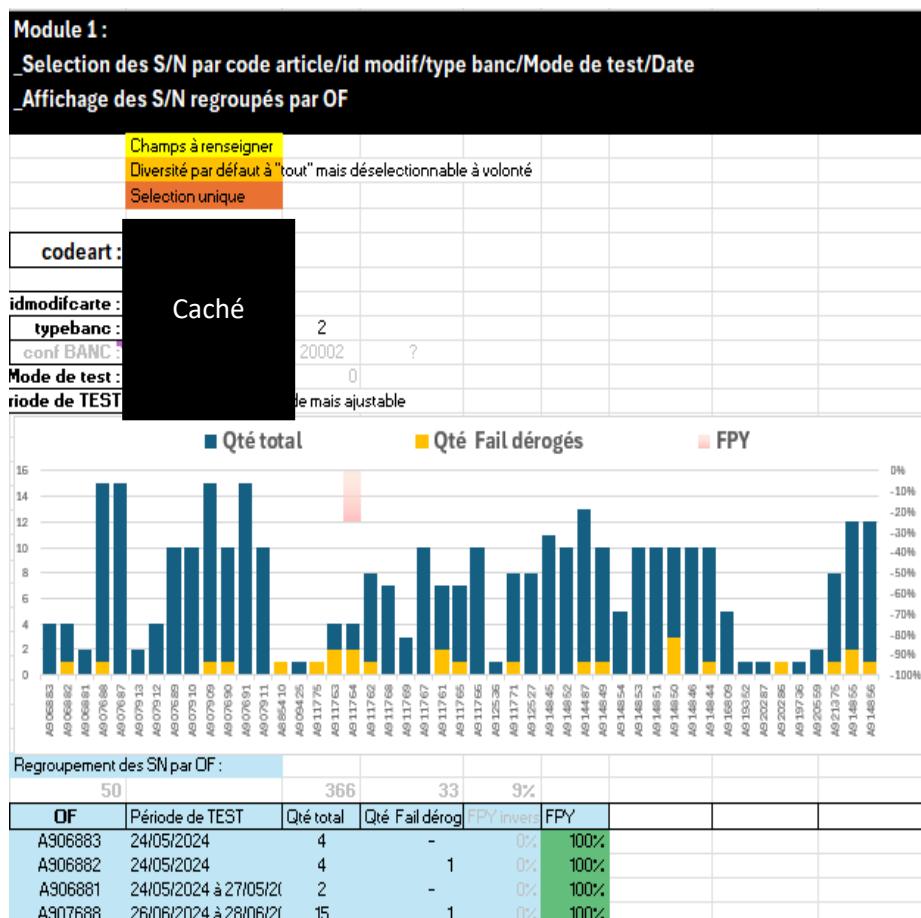


Figure 4 Exemple d'une partie de l'applicatif attendu.

2. Récupération des données en SQL

Une fois que j'ai pris connaissance de l'ensemble des documents, je me suis mis au travail pour commencer le développement du site web.

Dans un premier temps, j'ai eu besoin d'extraire les données destinées à alimenter les listes déroulantes présentes sur le site. Chaque liste déroulante dépendant de la sélection effectuée dans la précédente, il a fallu mettre en place un système de filtres progressifs dans les requêtes SQL, afin de ne récupérer que les données pertinentes. J'ai commencé par rédiger des requêtes simples, basées sur des instructions classiques telles que SELECT, FROM, INNER JOIN, WHERE, complétées si besoin par des clauses GROUP BY et ORDER BY.

Une fois les données de filtrage extraites, il ne me restait plus qu'à récupérer les données nécessaires à la création des différents graphiques.

Cependant, lors du déploiement du site sur les serveurs de l'entreprise, nous avons constaté un problème majeur : certaines requêtes mettaient un temps extrêmement long à s'exécuter, dépassant parfois 8000 secondes, alors qu'elles fonctionnaient correctement en local. Après investigation, nous avons identifié l'origine du problème : une différence de version de MySQL entre mon environnement local et celui des serveurs de l'entreprise, ces derniers utilisant une version beaucoup plus ancienne (j'avais une version 9.2 et la version sur leur serveur était en 5.6). Cela provoquait une forte latence dans le traitement des requêtes. Une demande a donc été faite pour mettre à jour la version de MySQL sur les serveurs.

En attendant cette mise à jour, j'ai cherché à optimiser mes requêtes. M. Chauchat m'a présenté une technique d'optimisation qui consiste à appliquer les filtres directement dans les clauses FROM ou INNER JOIN, afin de réduire la quantité de données à traiter en amont par le moteur SQL. Cette méthode repose sur la création de sous-requêtes dans ces clauses : on met une requête entre parenthèses, et on lui attribue un alias, et dans cette requête on sélectionne uniquement les données utiles tout en y appliquant directement les filtres via une clause WHERE.

Vous trouverez un exemple de ces requêtes avec les figures 5 et 6.

```

$query = "SELECT DISTINCT
    r.idtest,
    s.testname,
    s.min AS spec_min,
    s.max AS spec_max,
    s.unite,
    r.resultat,
    r.timestamp
FROM resultats r
INNER JOIN resultatglobal rg ON r.numserie = rg.numserie
INNER JOIN specifications s ON s.idtest = r.idtest
WHERE r.codeart = :codeart
    AND s.idmodifcarte IN ($inClause)
    AND r.modetest = :modetest
    AND s.typebanc = :typebanc
    AND (:mesver IS NULL OR CAST(s.mesver AS DECIMAL(5, 2)) = :mesver)
    AND rg.date BETWEEN :date_min AND :date_max
    order by idtest;

```

Figure 5 Ancienne requête

```

$query = "SELECT DISTINCT
    r.idtest,
    s.testname,
    s.min AS spec_min,
    s.max AS spec_max,
    s.unite,
    r.resultat,
    r.timestamp,
    r.status
FROM
    (select distinct idtest, resultat, timestamp, numserie, status
        from resultats
        where codeart = :codeart
            and modetest = :modetest
            and LEFT(verconfbanc, LENGTH(verconfbanc) - 4) = :typebanc) as r
INNER JOIN
    (select distinct date, numserie
        from resultatglobal
        where date between :date_min and :date_max) as rg
ON r.numserie = rg.numserie
INNER JOIN
    (select distinct testname, min, max, unite, idtest
        from specifications
        where idmodifcarte in ($inClause)
            and (:mesver IS NULL OR CAST(mesver AS DECIMAL(5, 2)) = :mesver)) as s
ON s.idtest = r.idtest
    order by idtest;

```

Figure 6 Nouvelle requête

3. Développement du back-end (PHP) et paramétrisation des requêtes

Une fois l'ensemble de mes requêtes SQL rédigées, il a fallu les intégrer dans des fichiers PHP afin de les rendre dynamiques et adaptables aux sélections effectuées par les utilisateurs dans les listes déroulantes. Comme mentionné précédemment, ces listes étaient toutes dépendantes les unes des autres, chaque filtre conditionnant les suivants.

Pour cela, j'ai d'abord récupéré les données envoyées dans l'URL par les fonctions JavaScript à l'aide de la méthode GET, puis je les ai assignées à des variables PHP. Ces variables me permettaient ensuite de personnaliser mes requêtes SQL en fonction des choix de l'utilisateur. Dans la clause WHERE de chaque requête, j'ai utilisé des paramètres nommés précédés du caractère : comme par exemple WHERE codeart = :codeart.

J'ai ensuite utilisé la méthode bindValue pour lier les paramètres de la requête aux valeurs récupérées via \$_GET. Une fois la requête exécutée, les données étaient renvoyées au format JSON, ce qui permettait à mes fichiers JavaScript de les décoder et de les utiliser pour générer dynamiquement les listes déroulantes, les tableaux ou encore les graphiques.

Dans certains cas, je ne renvoyais pas directement les données brutes dans le JSON. Je les traitais au préalable dans le fichier PHP pour effectuer différents calculs statistiques, comme le Cp, le Cpk ou d'autres indicateurs, puis je retournais uniquement les résultats de ces calculs dans la réponse JSON.

Le Cp et le Cpk étaient les deux indicateurs clés de ce projet. Ils commencent à être viables à partir d'une population de 30 individus en général. Pour mon code, nous avons commencé à tracer la courbe à partir de 3 individus, car certains articles n'ont pas suffisamment de résultats pour atteindre 30 individus. De plus, lorsque l'écart-type est nul, plutôt que de provoquer une erreur, le graphique n'affiche rien.

Le Cp (Capability Process) est un indicateur de capacité utilisé en contrôle qualité pour évaluer la capacité d'un processus de production à respecter les tolérances spécifiées.

Il mesure la dispersion des résultats par rapport aux limites de tolérance (USL = limite maximum et LSL = limite minimum), sans tenir compte du centrage des valeurs. On peut le calculer avec la formule (σ représente l'écart-type) :

$$CP = \frac{USL - LSL}{6\sigma}$$

Le Cpk (Capability Process Index) est un indicateur de capabilité qui, comme le Cp, mesure si un processus de production respecte les tolérances spécifiées, mais en prenant en compte le centrage du processus par rapport à la cible.

Il permet donc d'évaluer à la fois la dispersion et le décalage du processus par rapport aux limites de tolérance. On peut le calculer avec la formule (μ représente la moyenne des données) :

$$Cpk = \min\left(\frac{USL - \mu}{3\sigma}, \frac{\mu - LSL}{3\sigma}\right)$$

On peut voir comment j'ai intégré ces calculs dans mon code à la figure 7. Au préalable, j'avais développé une fonction stddev permettant de calculer l'écart-type.

```
// Si pas assez de données ou specs invalides, pas de calcul
if (count($all_values) < 3 || $spec_min === null || $spec_max === null) {
    $cp = null;
    $cpk = null;
    $cm = null;
} else {
    $mean = array_sum($all_values) / count($all_values);
    $std = stddev($all_values);
    if ($std == 0) $std = 1e-10;// évite division par 0

    // Calcul des indices de capabilité
    $cp = ($spec_max - $spec_min) / (6 * $std);
    $cpk = min(
        ($spec_max - $mean) / (3 * $std),
        ($mean - $spec_min) / (3 * $std)
    );
    $cm = $cp / (1 + 9 * pow($cp - $cpk, 2));
}
```

Figure 7 Intégration du calcul des Cp/Cpk dans mon code

4. Conception de l'interface utilisateur (front-end)

La partie principale du travail a consisté à développer l'interface du site web et à mettre en place l'ensemble des mécanismes dynamiques permettant l'affichage des données.

J'ai commencé par concevoir la structure de la page web en HTML. C'est à ce moment-là que j'ai défini les emplacements de tous les éléments nécessaires : listes déroulantes, tableaux et graphiques.

À ce code HTML, j'ai associé des fichiers CSS, afin d'améliorer le rendu visuel de la page et de veiller à ce que les formulaires, tableaux et graphiques soient correctement alignés et lisibles.

Une fois la structure en place, il a fallu remplir dynamiquement les différents éléments avec les données issues de la base. Pour cela, j'ai utilisé plusieurs fonctions JavaScript, reposant toutes sur un fonctionnement commun.

Tout d'abord, j'ai créé des variables globales pour stocker les données sélectionnées dans les listes déroulantes, en utilisant `document.getElementById`. Ces variables permettaient de récupérer les valeurs choisies par l'utilisateur.

```
const codeartSelect = document.getElementById("codeart"); // Le menu déroulant pour choisir un code article
const typebancSelect = document.getElementById("typebanc"); // Le menu pour choisir un type de banc
const modetestSelect = document.getElementById("modetest"); // Le menu pour choisir un mode de test
const dateDebut = document.getElementById("date_debut"); // Le champ pour la date de début
const dateFin = document.getElementById("date_fin"); // Le champ pour la date de fin
const idmodifContainer = document.getElementById("idmodifcarte_container"); // Le conteneur qui affiche ou cache les cases à cocher
const idmodifcheckboxes = document.getElementById("idmodifcarte_checkboxes"); // La zone où seront placées les cases à cocher
const mesverSelect = document.getElementById("mesver"); // Le menu déroulant pour choisir une version de MES
const snSelect = document.getElementById("SN"); // Le menu pour choisir un numéro de série (SN = Serial Number)
let currentIdtest = null;
```

Figure 8 Définition des variables globales

Ensuite, j'ai construit un objet `params` qui servait à transmettre les paramètres dans l'URL vers les fichiers PHP. Ces paramètres étaient ajoutés à l'aide de la méthode `params.append`.

J'ai ensuite utilisé `fetch` pour envoyer la requête au fichier PHP correspondant, et récupéré la réponse au format JSON grâce à `response.json()`.

```
// Etape 2 : On prépare les paramètres pour l'URL
const params = new URLSearchParams();
params.append("codeart", codeart);
idmodifcarte.forEach(id => params.append("idmodifcarte[]", id)); // Ajouter chaque idmodifcarte
if (typebanc) params.append("typebanc", typebanc); // Type de banc si sélectionné
if (modetestVal) params.append("modetest", modetestVal); // Mode de test si sélectionné
// On récupère aussi un éventuel filtre de statut (par ex : "only fails")
if (statusVal) params.append("status", statusVal);
const showNbFail = statusVal === "0" || statusVal === "all"; // Si statut est "0" (fail uniquement), on ajoutera une colonne spéciale "Nb Fail"
// Ajout des dates si elles existent
if (dateDebutVal) params.append("date_debut", dateDebutVal);
if (dateFinVal) params.append("date_fin", dateFinVal);
if (!mesverList?.[mesverVal]?.isCustom) {
  params.append("mesver", mesverVal);
} // Ajout de la version MES
if (snVal) params.append("SN", snVal); // Ajout du numéro de série si sélectionné
const showfpy = snVal !== "0"; // on affichera la colonne fpy si un numéro de série est sélectionné.
startLoading();
try {
  // Etape 3 : On envoie la requête pour récupérer les spécifications de tests
  const specResponse = await fetch("api/objects/spec.php" + params.toString());
  const specData = await specResponse.json(); // Réponse transformée en tableau d'objets
```

Figure 9 Exemple de `params.append` et de `fetch`

Prenons l'exemple de la création d'un tableau. Une fois les données reçues, j'ai ciblé la balise `<tbody>` du tableau en créant une variable dédiée. À chaque exécution de la fonction, j'ai vidé le contenu du tableau en initialisant `tbody.innerHTML = ""`, puis je l'ai rempli à nouveau en construisant dynamiquement des lignes HTML contenant les valeurs extraites précédemment.

Enfin, pour que ces actions se déclenchent au bon moment, j'ai ajouté des EventListener sur les listes déroulantes. Ces événements permettaient de lancer la fonction de création du tableau une fois que l'utilisateur avait sélectionné le dernier filtre nécessaire.

```
// -----
// Changement du mode de test
// -----
modetestSelect.addEventListener("change", () => {
    fetchPeriodetest(); // Met à jour la période de test
    fetchMesver(); // Met à jour les mesures
    setTimeout(() => {
        updateFpyTable();
        updateFpyGraph();
    }, 1000);
    clearTable();
});
```

Figure 10 Exemple d'un EventListener

En parallèle du développement principal, j'ai également mis en place deux petites fonctions JavaScript destinées à gérer une roue de chargement animée, conçue en CSS. Celle-ci s'affiche pendant les temps de traitement et disparaît une fois les données chargées, afin d'améliorer l'expérience utilisateur.

Au cours de ce stage, j'ai également anticipé une partie du programme de troisième année de BUT en apprenant à créer des graphiques en JavaScript. Pour cela, j'ai utilisé la bibliothèque Chart.js, que j'ai intégrée à ma page HTML en l'appelant dans la balise <head> via :

```
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
```

Sur la page, j'ai placé plusieurs balises <canvas> avec des id différents, afin d'indiquer à JavaScript où dessiner chaque graphique.

Le fonctionnement général pour générer les graphiques est similaire à celui des tableaux. Je commence par récupérer les choix de l'utilisateur via les éléments du formulaire, puis je transmets ces données aux fichiers PHP à l'aide de URLSearchParams() et d'un fetch(). La réponse est ensuite récupérée au format JSON.

Avant d'afficher le graphique, je déclare les variables nécessaires et effectue les calculs préparatoires, comme les Cp ou Cpk selon les cas.

Ensuite, je cible l'élément <canvas> correspondant avec une instruction de ce type :

```
const ctx = document.getElementById("CPKChart").getContext("2d");
```

Pour éviter tout problème d'affichage lors de l'actualisation des filtres, j'ai mis en place un mécanisme de suppression du graphique existant avant d'en créer un nouveau :

```
if (window.cpkChart && typeof window.cpkChart.destroy === 'function') {  
    window.cpkChart.destroy();}
```

Cela garantit que le graphique soit toujours à jour, sans superposition ni doublon lors des rafraîchissements.

Pour créer un graphique avec Chart.js, il faut déclarer une variable à laquelle on associe une instance de la fonction Chart. Cette fonction prend deux paramètres : la variable ctx, créée précédemment pour cibler le <canvas>, et un objet contenant l'ensemble des options de configuration du graphique.

La première étape consiste à définir le type de graphique souhaité. Dans mon cas, j'ai principalement utilisé des graphiques en lignes (line) et en barres (bar). Une fois le type défini, il faut configurer l'affichage des données, c'est-à-dire définir les courbes ou les barres à tracer.

Cela passe par la déclaration d'un ou plusieurs ensembles de données (datasets), dans lesquels on indique les valeurs à représenter (data), ainsi qu'un label descriptif pour la légende du graphique. On peut également personnaliser l'apparence en ajoutant différentes options : couleur de la ligne ou des barres, affichage ou non des points, taille des points, épaisseur des lignes, etc. Si l'on souhaite représenter plusieurs axes Y sur un même graphique, c'est également dans cette configuration que l'on précise à quel axe se réfère chaque courbe. Pour cela, il suffit d'ajouter la propriété yAxisID dans l'objet de la courbe concernée, et de l'associer à l'identifiant de l'axe défini dans la section des options. Cela permet, par exemple, d'afficher simultanément deux courbes ayant des unités ou des ordres de grandeur différents, tout en maintenant une lisibilité optimale. Un exemple de cette partie de la création de graphique est disponible sur la figure 11.

Toutes ces options sont structurées sous forme d'un objet JavaScript, suivant un format clé/valeur entre accolades, similaire à un dictionnaire.

```
// Cr ation du graphique Chart.js avec les datasets CP / CPK et les options
window.cpkChart = new Chart(ctx, {
    type: "line",
    data: {
        labels: rawLabels, //  tiquettes sur l'axe X
        datasets: [
            // Courbe CP
            {
                label: "CP",
                data: rawLabels.map((l, i) => ({ x: l, y: cpArray[i] })),
                borderColor: "#2c3e50",
                backgroundColor: "transparent",
                tension: 0.2,
                pointRadius: 0,
                order: 1
            },
            // Courbe CPK
            {
                label: "CPK",
                data: rawLabels.map((l, i) => ({ x: l, y: cpkArray[i] })),
                borderColor: "red",
                backgroundColor: "transparent",
                tension: 0.2,
                pointRadius: 0,
                order: 1
            }
        ]
});
```

Figure 11 Cr ation d'un graphique et insertion des valeurs

Une fois les courbes cr  es, il est  galement possible d'ajouter diff  entes options pour personnaliser et am  liorer l'affichage des graphiques.

Parmi les plus utiles, l'option responsive permet au graphique de s'adapter automatiquement   la taille de l' cran. Cela est particuli  rement pratique lorsque l'utilisateur utilise un ordinateur portable ou un  cran de petite taille, car cela garantit une lecture optimale du contenu.

On peut  galement activer l'option interaction, qui permet d'afficher des informations d taill es (comme les valeurs exactes d'un point) au survol de la souris, am  liorant ainsi l'interpr tation des donn es.

Enfin, l'une des options les plus importantes concerne la personnalisation des axes. Il est en effet possible de modifier les axes X et Y afin d'adapter le graphique au type de donn es affich es. Cela permet, par exemple, d'associer des noms explicites aux points de l'axe des abscisses (codes articles, dates, identifiants de test), ou encore d'ajouter un axe secondaire   droite du graphique. Ces r glages se font dans la section options du graphique, o  il est possible de configurer les libell s, les positions, les  chelles ou encore les unit es associ es   chaque axe. Cela permet de rendre la visualisation plus claire, plus lisible et mieux adapt e aux besoins des utilisateurs.

```

options: {
    responsive: true, // Adapté aux différentes tailles d'écrans
    maintainAspectRatio: false,
    scales: {
        x: {
            stacked: true, // Empiler les barres sur l'axe X
        },
        y: {
            stacked: true, // Empiler les barres sur l'axe Y principal
            position: 'left', // L'axe Y est à gauche
            beginAtZero: true, // Commencer à 0
            title: {
                display: true,
                text: 'Quantités' // Titre pour les quantités
            }
        },
        y1: {
            position: 'right', // L'axe Y1 est à droite
            beginAtZero: true, // Commencer à 0
            min: 0, // L'échelle va de -100 à 0 pour le FPY
            max: 100,
            title: {
                display: true,
                text: 'FPY (%)' // Titre de l'axe Y1
            },
            grid: {
                drawOnChartArea: false // Ne pas dessiner la grille pour l'axe secondaire
            },
            ticks: {
                // Formate les valeurs affichées pour ajouter le signe "%"
                callback: value => value + '%'
            }
        }
    }
},

```

Figure 12 Modification des options et axes du graphique

5. Présentation du résultat

Le produit fini est divisé en trois modules distincts.

Le premier module permet à l'utilisateur de sélectionner un code article ainsi que plusieurs autres filtres complémentaires. L'objectif est ensuite de regrouper les numéros de série correspondant à ces critères, en les classant par ordre de fabrication, puis de calculer le FPY (First Pass Yield) à partir de ces données.

Les résultats sont d'abord présentés sous forme de tableau, trié chronologiquement, puis visualisés à l'aide d'un graphique. Cela permet à l'utilisateur d'identifier rapidement les tendances ou les anomalies de production.

Le rendu de ce module dans sa version web est présenté aux figures 13 et 14.

CONTROLE PROCESS



Figure 13 Tableau FPY



Figure 14 Graphique FPY

Le deuxième module reprend les filtres utilisés dans le premier module, tout en y ajoutant trois filtres supplémentaires afin de permettre une sélection plus fine des données.

L'objectif de ce module est de fournir un tableau détaillé affichant des résultats précis pour chaque test réalisé. Parmi les indicateurs les plus importants de ce tableau figurent les calculs des indices Cp et Cpk, qui permettent d'évaluer la capabilité du processus de production.

Une fonctionnalité clé de ce module réside dans le fait que les identifiants de test (ID de test) sont présentés sous forme de boutons cliquables. En cliquant sur l'un de ces boutons, l'utilisateur peut sélectionner le test spécifique à analyser dans le troisième module, dédié à la génération de graphiques.

L'utilisateur a également la possibilité de créer une version de mesure personnalisée en modifiant les bornes de spécifications minimales et maximales directement dans

le tableau. Cette modification a un impact direct sur le calcul des indices Cp et Cpk, permettant ainsi d'adapter les critères de test en fonction des résultats reçus depuis les bancs de test.

Nous avons également mis en place un code couleur pour visualiser rapidement les résultats des indices Cp et Cpk :

- Rouge : le Cp ou le Cpk est inférieur à 1. Cela signifie que la dispersion des mesures n'est pas acceptable, car le taux de non-conformité prédictif est supérieur à 0,27 %. Dans ce cas, les critères de test ou le produit ne sont pas conformes aux attentes QCD.
- Orange : le Cp ou le Cpk est compris entre 1 et 1,33. La dispersion des mesures est considérée comme acceptable sous dérogation, car le taux prédictif de non-conformité est compris entre 0,27 % et 0,0066 %.
- Vert : le Cp et le Cpk sont supérieurs ou égaux à 1,33. La dispersion des mesures est considérée acceptable. Le test est conforme.
- Gris : le Cp ou le Cpk n'a pas pu être calculé, généralement à cause d'un écart-type nul.
- Violet : une erreur dans les données de calcul (par exemple une borne minimale supérieure à la borne maximale ou une mesure aberrante qui donne une moyenne incohérente) a conduit à un résultat incohérent ou négatif.

Ce système permet une lecture rapide de la qualité des tests et aide à identifier immédiatement les cas problématiques.

The screenshot displays a software application for managing measurement data. At the top, there's a header bar with buttons for 'Nom de Mesur', 'ID Test', 'Spec Min', 'Spec Max', and 'AJOUTER UNE VERSION DE MES'. Below this is a table with columns: idtest, Nom du test, NB de résultats, Spec min, Spec max, unité, Mesure min, Mesure max, Moy, Nb Fail, CP, CPK, and Écart-type (σ). The table lists several test entries, each with a status indicator (color box) next to the CP and CPK values. A legend on the left side provides a key for these colors:

- Incohérences sur les bornes (purple)
- Incapable de calculer Cp/Cpk (écart-type nul par exemple) (grey)
- Cp/Cpk NOK (red)
- Cp/Cpk avec dérogation (orange)
- Cp/Cpk OK (green)

idtest	Nom du test	NB de résultats	Spec min	Spec max	unité	Mesure min	Mesure max	Moy	Nb Fail	CP	CPK	Écart-type (σ)
10020	Conso 24V TEK1	437	90	120	mA	0	237	110.460	34	0.27	0.17	18.6509
10030	Alimentation 24 Volts	422	23.7	24.3	V	0.0002888999879360199	23.98875617980957	23.924	1	0.09	0.06	1.1674
	Alimentation 18 Volts	421	17.3	18.7	V	17.504276275634766	18.121414194570312	17.825	0	3.45	2.59	0.0675
	Alimentation 15V-1	421	14.5	15.5	V	14.89630544311523	15.283607591562734	15.074	0	2.39	2.04	0.0697
	Alimentation 15V-2	421	14.5	15.5	V	14.88102912902832	15.253583908081055	15.074	0	2.50	2.13	0.0667
	Alimentation 5 Volts	421	4.95	5.15	V	4.950672523498535	5.048478603363037	5.006	0	1.70	0.94	0.0196
	Alimentation 3.3 Volts	421	3.1	3.5	V	3.272714853206743	3.32877516746521	3.301	0	4.93	4.90	0.0136
	Sortie VR2 Alim. 31V	421	17.3	18.7	Volts	17.506580352783203	18.121692657470703	17.828	0	3.48	2.63	0.067
	Commutation PLL1	418	3.135	3.465	KOhm	2.7266268730163574	9.900000603163346e+34	1.184210598464515e+34	8	0.00	-0.04	1.0775540925108548e+34
	Commutation PLL2	408	5.32	5.88	KOhm	4.083394527435303	5.6286540031433105	5.610	1	1.23	1.19	0.0759
	Commutation PLL3	407	14.25	15.75	KOhm	14.97964572964941	15.054924964904785	15.017	0	18.74	18.32	0.0133
	R12/R11	407	38	42	KOhm	40.36039352416992	40.660743713378906	40.497	0	13.60	10.22	0.049
	Commutation PLL4	406	95	105	KOhm	99.52777099609375	100.76000213623047	100.092	0	7.18	7.05	0.2322

Figure 15 Tableau de résultats avec calculs des Cp/Cpk

Pour finir, le troisième module consistait à générer quatre graphiques à partir des résultats associés à un identifiant de test sélectionné dans le tableau du module précédent.

Le premier graphique est une droite de Henry. Il permet de visualiser la répartition des résultats autour de cette droite de référence. Si les points sont globalement alignés et répartis de manière équilibrée autour de la droite, cela signifie que l'échantillon suit une loi normale. Dans le cas contraire, on peut conclure que la distribution des données s'écarte de la normalité, ce qui signifie que l'approche statistique des indicateurs Cp et Cpk n'est pas pertinente.

Le deuxième graphique représente les résultats par rapport à une courbe théorique de loi normale. Il permet d'évaluer la distribution pour un grand nombre d'échantillons, et donc de prédire les quantités dépassant les limites.

Le troisième graphique affiche l'évolution des résultats du test sélectionné pour l'ensemble des numéros de série d'un code article donné, au fil du temps. Il permet d'identifier d'éventuelles dérives ou anomalies dans les résultats, possiblement liées à la température ambiante ou à la configuration du test.

Enfin, le quatrième graphique, lié au précédent, représente les indicateurs Cp et Cpk de ces mêmes numéros de série au cours du temps. Ce graphique permet de suivre l'évolution de la capabilité du processus et de détecter toute dérive dans la production.

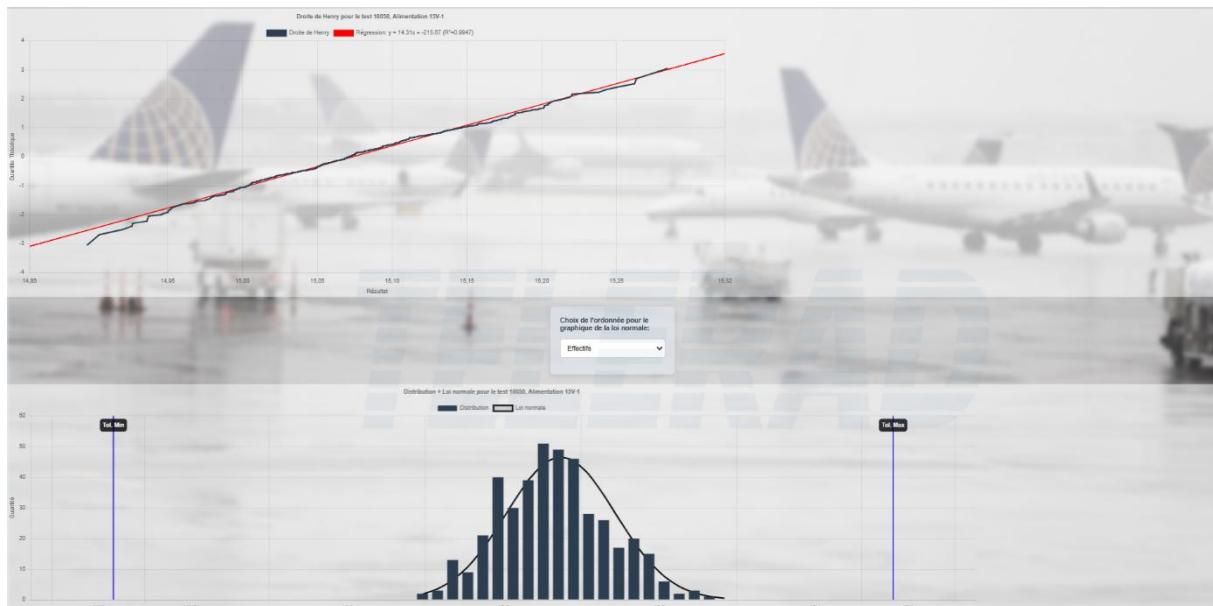


Figure 16 Droite de Henry et distribution de loi normale

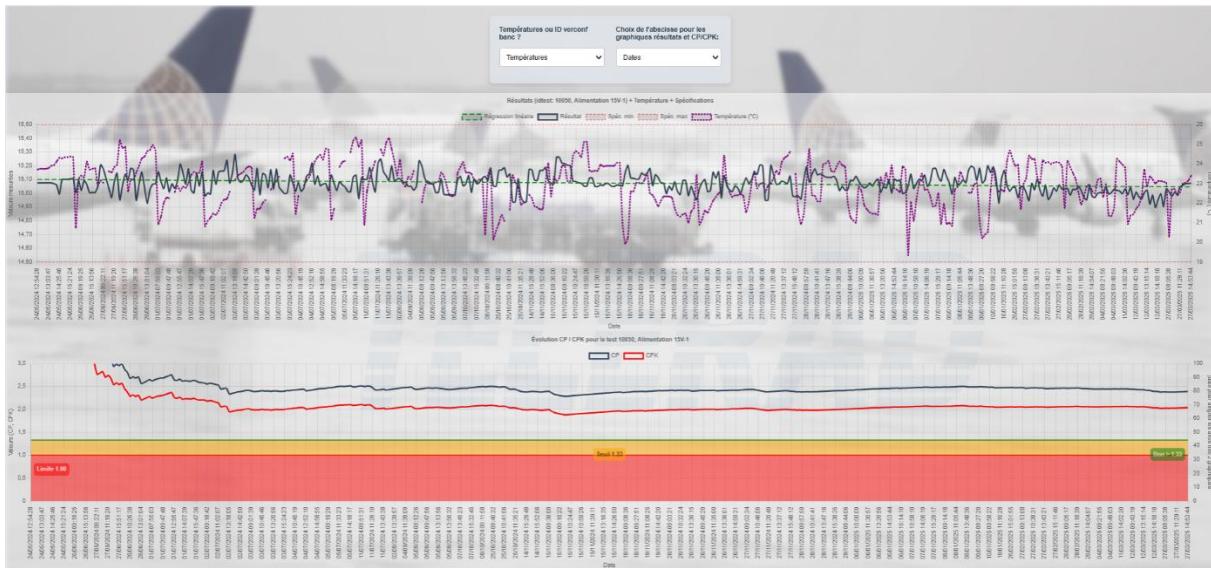


Figure 17 graphique des résultats et courbes Cp/Cpk

Conclusion

Ce stage m'a permis de découvrir concrètement le développement web en environnement professionnel, et de renforcer mes compétences dans ce domaine. J'ai ainsi pu appréhender les exigences techniques, les méthodes de travail en autonomie et l'importance de proposer des solutions adaptées aux besoins d'un service métier.

Cette expérience a également élargi mes perspectives professionnelles, en me faisant découvrir un domaine vers lequel je ne m'étais pas particulièrement tourné au départ, mais qui s'est révélé enrichissant et stimulant.

Ce stage représente une étape importante dans mon parcours. Je souhaite poursuivre mon évolution l'année prochaine dans le cadre d'une alternance, idéalement orientée vers le domaine de l'analyse de données sportives, avec l'objectif d'occuper un poste de data analyst, notamment au sein d'un club de rugby.

Bilan personnel

Ce stage a été une très bonne expérience, tant sur le plan technique que personnel. Il m'a permis de progresser significativement en programmation, notamment en JavaScript, PHP et SQL, des langages dans lesquels je ressentais auparavant certaines lacunes.

J'ai également appris à mieux me connaître : je me suis rendu compte que je me mettais souvent beaucoup de pression pour des choses que je suis finalement capable de gérer. Avant le début du stage, l'idée de devoir développer un site web me faisait paniquer, car je ne me sentais pas à la hauteur. Pourtant, après seulement quelques jours, j'ai réussi à mettre en place les premières fonctionnalités sans difficulté particulière. Cela m'a donné confiance en mes capacités.

J'ai aussi constaté que je savais m'organiser rapidement et de manière efficace, ce qui a facilité le bon déroulement du projet.

Avant ce stage, je n'étais pas particulièrement attiré par la programmation web. Aujourd'hui, même si ce n'est pas le domaine qui me passionne le plus, je ne ferme pas la porte à la possibilité d'y faire carrière. Le fait d'avoir pu expérimenter ce travail dans un contexte professionnel a changé ma perception et ouvert de nouvelles perspectives.

Glossaire

Termes	Définitions
HTML	<p><u>HTML (HyperText Markup Language)</u> est le langage de balisage utilisé pour structurer le contenu d'une page web. Il permet de définir les éléments d'une page comme les titres, les paragraphes, les images... Il constitue la base de toute page web.</p>
CSS	<p><u>CSS (Cascading Style Sheets)</u> est un langage utilisé pour décrire l'apparence et la mise en forme des pages web écrites en HTML. Il permet de définir le style des éléments (couleurs, polices, tailles, marges, positionnement, etc.) et de rendre les pages plus esthétiques et agréables à utiliser.</p>
JavaScript	<p><u>JS (JavaScript)</u> est un langage de programmation utilisé pour rendre les pages web interactives et dynamiques. Il permet, par exemple, de gérer des événements (clics, saisies, survols), de modifier le contenu d'une page sans la recharger, ou encore de créer des animations, des graphiques, et des formulaires interactifs. Il s'exécute directement dans le navigateur de l'utilisateur.</p>
PHP	<p><u>PHP (Hypertext Preprocessor)</u> est un langage de programmation côté serveur utilisé pour créer des pages web dynamiques. Il permet de générer du contenu HTML en fonction des actions de l'utilisateur, de traiter des formulaires, de gérer des sessions, et surtout d'interagir avec des bases de données. Le code PHP est exécuté sur le serveur, puis le résultat est envoyé au navigateur de l'utilisateur.</p>

SQL	<p><u>SQL (Structured Query Language)</u> est un langage utilisé pour interagir avec les bases de données. Il permet de créer, modifier, interroger et supprimer des données stockées dans des bases. Grâce à SQL, on peut récupérer des informations précises à l'aide de requêtes, filtrer les résultats, faire des calculs ou encore lier plusieurs tables entre elles.</p>
FPY	<p><u>FPY (First Pass Yield)</u>, ou rendement au premier passage, est un indicateur de qualité utilisé en production. Il mesure le pourcentage de produits qui passent avec succès toutes les étapes de test ou de fabrication du premier coup, sans nécessiter de retouche ou de reprise.</p>
CP	<p><u>Cp (Capability Process)</u> est un indicateur de capacité utilisé en contrôle qualité pour évaluer la capacité d'un processus de production à respecter les tolérances spécifiées.</p> <p>Il mesure la dispersion des résultats par rapport aux limites de tolérance, sans tenir compte du centrage des valeurs.</p>
CPK	<p><u>Cpk (Capability Process Index)</u> est un indicateur de capacité qui, comme le Cp, mesure si un processus de production respecte les tolérances spécifiées, mais en prenant en compte le centrage du processus par rapport à la cible.</p> <p>Il permet donc d'évaluer à la fois la dispersion et le décalage du processus par rapport aux limites de tolérance.</p>

Table des annexes :

Annexe 1 Description de la base de données	30
Annexe 2 Description d'une table de la base de données	31

Caché

Caché

Annexe 2 Description d'une table de la base de données